

CONCURRENT PROCESSING SIMULATION OF THE SPACE STATION

by

R. Gluck, TRW Space & Technology Group, Redondo Beach, CA. 90278
A. L. Hale, Supercomputing Solutions Inc., San Diego, CA. 92121
J. W. Sunkel, NASA Johnson space Center, Houston, TX., 77058

ABSTRACT

This report describes the development of a new capability for the time-domain simulation of multibody dynamic systems and its application to the study of a large-angle rotational maneuvers of the Space Station. The effort was divided into three sequential tasks, which required significant advancements of the state-of-the art to accomplish. These were: a) the development of an explicit mathematical model via symbol manipulation of a flexible, multibody dynamic system; b) the development of a methodology for balancing the computational load of an explicit mathematical model for concurrent processing, and c) the implementation and successful simulation of the above on a prototype Custom Architected Parallel Processing System (CAPPS) containing eight processors.

The throughput rate achieved by the CAPPS operating at only 70 percent efficiency, was 3.9 times greater than that obtained sequentially by the IBM 3090 supercomputer simulating the same problem. More significantly, analysis of the results leads to the conclusion that the relative cost-effectiveness of concurrent vs. sequential digital computation will grow substantially as the computational load is increased. This is a welcomed development in an era when very complex and cumbersome mathematical models of large space vehicles must be used as substitutes for full-scale testing which has become impractical.

1.0 INTRODUCTION

The Space Station exemplifies future NASA missions, which contemplate the use of large, flexible multibody space vehicles requiring structural dynamics control to meet their objectives. Because of their large size and limberness, full scale development and verification testing of these vehicles in the laboratory is impractical. Even if such tests could be made, results obtained in the earth gravitational environment are often misleading or inconclusive regarding the vehicle's on-orbit behavior. For these reasons, analytical modeling and simulation have become essential tools for large space structures design.

To satisfy the designer's needs, analytical modeling and simulation tools for large space structures must possess the following attributes:

- Accommodate all desired rigid-and flexible-body degrees of freedom of the system and incorporate acceptable models of its control system(s) and external forces and torques acting on it.
- Require short computation times and keep computation costs within reasonable bounds.
- Are versatile enough to accommodate radical variations in space structure configuration from one study to the next.

The most readily available analytical simulation tools in the aerospace industry are sequential digital computers. The most common among these are large mainframe computers and supercomputers which do meet high fidelity and versatility requirements, but only with a crippling penalty of simulation time and cost. Moreover, experience gathered at TRW over the past several years strongly suggests that the execution speed of conventionally coded software on commercially available sequential computers is rapidly approaching a limit; only relatively modest improvements in simulation throughput rate can be expected for these computers in the near future. Yet, the cost-per-run, at present, for even the most efficient of them is excessive and precludes comprehensive simulation studies or meaningful support of the design process.

This paper describes the results of a project undertaken to demonstrate the application of a specific concurrent processing system, the Custom Architected Parallel Processing System (CAPPS), in determining the control/structure interaction of a representative Space Station undergoing a large angle maneuver. The project was carried out under a NASA contract (NAS 9-17778) with the Johnson Space Center. It consisted of the following three tasks:

- (a) Develop an explicit control/structure interaction model of the Space Station. This task was a joint effort of TRW and NASA personnel, the latter providing the structural data and control models and the former applying these data to the development of an explicit mathematical model of the Space Station via symbol manipulation.
- (b) Distribute the computational load for the CAPPS. A methodology for a balanced computational load distribution was applied to the Space Station model of Task (a) to prepare it for concurrent processing on the CAPPS.

- (c) Demonstrate the CAPPS multiprocessor. In this task, the control/structure interaction of the Space Station model was simulated using a CAPPS containing 8 Computational Units (processors). The simulation speedup achieved by this concurrent processor was measured and compared to the performance of sequential digital computers simulating the same problem.

This paper is divided into 5 sections. Sections 2, 3 and 4 are devoted to the work accomplished under Tasks (a), (b) and (c), respectively. Section 5 contains the conclusions drawn from the results obtained. Further details of the Space Station simulation and CAPPS implementation are contained in Reference 1.

2.0 SPACE STATION MODEL DEVELOPMENT

2.1 Derivation of the Equations of Motion

A non-linear mathematical model describing the fully coupled rigid-and flexible-body motion of the Space Station undergoing a large angle maneuver was derived in explicit (scalar) mathematical form using Kane's dynamical equations. Explicit equations provide the analyst with considerable engineering insight into the problem being solved, permitting fine tuning of the mathematical model, including the elimination of superfluous operations, such as additions of zeros, multiplications by unity, or the computations of dot products of orthogonal vectors. Moreover, the derivation of explicit dynamical equations of motion is performed only once, in contrast with conventional implicit formulations (such as Programs DISCOS and Treetops, References 2 and 3, respectively) in which the equations of motion are essentially rederived at each time step of the numerical integration. This leads to a significant reduction in simulation time of explicit models compared to implicit ones. In one example, a 4-fold increase in simulation speed was realized at TRW by an explicit model compared to that obtained with Program DISCOS simulating the same problem. Another advantage of explicit models is the ability to determine the degree of accuracy to which important parameters must be known to achieve a desired accuracy of the solution. Finally, explicit equations lend themselves well to "coarse grain" computational load distribution in preparation for concurrent processing simulation, as described in Section 3.

Explicit equations of motion are developed by applying the Symbolic Manipulation Program (SMP, see Reference 4) to the Space Station model. This method of generating explicit equations of motion in SMP using Kane's formulation will be hereafter designated as Program SYMBOD (Symbolic Multi-Body).

Program SYMBOD generates a set of ordinary differential equations of the form: $A(q,t)u = b(q,u,t)$, $q_d = f(q,u,t)$, where q and q_d are generalized coordinates and their first time derivatives, respectively, u and u_d are, respectively, generalized speeds and their first time derivatives, and t is time. Elements of A , b , and f are generated by SYMBOD and then translated into FORTRAN via file. Symbolically deriving the model eliminates the many coding errors and debugging steps required when equations of motion are formulated implicitly.

Developing an operational symbol manipulation methodology for deriving Kane's dynamical equations requires a systematic method of reducing the number of algebraic operations in the formulation of these equations. Frequently the intermediate computations of expressions, such as velocity terms, produce expressions so large that their storage requirements exceed the computer's capacity. Therefore, a procedure for systematically introducing new intermediate symbols to replace recurring combinations of algebraic subexpressions was developed in SYMBOD. This procedure eliminates repetitious calculations and results in efficient computational algorithms requiring fewer arithmetic operations and a vastly reduced computer storage.

A series of utility procedures were developed to generate symbolic expressions for partial velocities, partial angular velocities, their associated time derivatives, and the equations of motion. One important advantage of this novel approach of formulating the equations of motion is the analyst's ability to redefine quantities such as generalized speeds and partial velocities to fit his needs. This can be done very easily with just minor modifications to Program SYMBOD. In contrast, these revisions would require such a major modification in a conventional implicit formulation code, often making it impractical to accomplish. This very desirable feature is not available in any other simulation code for multibody dynamic systems. Its application, however, requires intensive interaction of an experienced analyst well versed in Kane's formalism.

2.2 Model Description

The physical system of the Space Station was described by three flexible bodies interconnected at the two ALPHA gimbals (or hinges) to form the topological tree configuration of Figure 1. The main central body (Body 1), containing the pressurized modules inboard of the two ALPHA gimbals, was selected as the reference body for the Space Station model. The starboard body (Body 2) and the port body (Body 3) each consisted of all the components, including the solar arrays, on the transverse boom outboard of the ALPHA gimbals.

Finite element models were developed for each body of the Space Station. They consisted of an unconstrained (free-free) model of the central body and two constrained (fixed-free) models of the starboard and port bodies cantilevered at the ALPHA gimbals. The characteristics of the finite element models are shown in Table 1. The MSC/NASTRAN program was used to obtain the natural modes of vibration within a 10.0 Hz frequency band. The spectrum of natural frequencies for each of the three finite element models is shown in Figure 2. Note that these are characterized by a number of low frequency modes (below 1 Hz) spaced closely together. Each of the bodies in the model was described by its own assumed admissible spatial functions which were extracted from the modal data.

The three-body Space Station model contained eight (8) large-motion, rigid-body degrees-of-freedom (dof), three translational and three rotational for the central body, and one rotational for each of the extraneous bodies relative to the central body. Full coupling between the rigid-and flexible-body dof was facilitated in the model. The flexibility of Body 1 was described by 44 "free-free" natural modes used here as assumed admissible functions. The flexibilities of Bodies 2 and 3 were each described by 44 "fixed-free" natural modes serving also as assumed admissible functions. The entire model consisted of 140 coupled rigid-and flexible-body dof.

The Space Station model was used to simulate a transient maneuver involving a large-angle, rigid-body rotation of the flexible solar arrays connected to the transverse booms, while maintaining the central body in a three-axis attitude control mode. Two separate control systems were incorporated in the model to simulate this maneuver. The first one was a three-axis attitude control system using uncoupled proportional-differential feedback control laws, designed to regulate the Space Station orientation and keep a longitudinal axis of the central body aligned with the local vertical, while maintaining a plane containing this axis perpendicular to the velocity vector. The control system consisted of attitude sensing instrumentation, control moment gyros, and electronics to cause corrective control moments to be applied to the Space Station central body whenever it moved away from the commanded attitude. The attitude rate sensors and the control moment gyros were co-located at the central body's undeformed center of mass.

The second control system executes the large-angle rotations of the ALPHA gimbals. This control system was designed to maintain the solar arrays pointing in a direction perpendicular to the sun line. The second order control law uses angular position and rate feedback of the ALPHA gimbal to calculate the controller's motor torque. Options were provided in the control law to rewind the solar arrays during eclipse. This control system was activated by rotating the spacecraft-sun line a specified angle away from the solar array's normal.

3.0 COMPUTATIONAL LOAD DISTRIBUTION

The optimization of a concurrent processor performance is achieved by minimizing that part of the computational load which must be performed sequentially. The realization of this statement, often identified as Amdahl's Law, is what makes the computational load distribution for concurrent processing a formidable task.

The explicit first-order Kane's equations of motion are integrated numerically using a fourth order Adams-Bashforth algorithm. This involves evaluating new u and q vectors at each time step based on computed values of u_d and q_d at the current and 3 preceding time steps. Evaluating the current u_d and q_d vectors, the derivative evaluation phase is based on computed values of u and q at the previous time step as well as t .

The derivative evaluation and numerical integration for the Space Station model were distributed among 8 CAPPS processors based on a "coarse-grain" decomposition of the data. Guided by the problem physics, the 8 rigid-body dof were allocated to processor 1, and 22 of the 44 flexible-body dof's per body were allocated to processors 2, 3, 5, 6, 7, and 8, which were paired so that processors 2 and 3 were dedicated to body 1, processors 5 and 6 to body 2, and processors 7 and 8 to body 3. Processor 4 was allocated computation associated with the coupling of bodies 2 and 3 to body 1, but it was not allocated any dof. Both computation and communication "costs" were considered carefully before choosing this distribution.

The computations for evaluating u_d and q_d at each time step, which are sequential for sequential execution, were next divided into numerous subroutines appropriate for the concurrent computation. Finally, the subroutines were distributed among the processors and communication of data was added as shown in Figure 3. The arrows in the figure show communication among the processors. The distribution is heterogeneous, i.e., different processors execute quite different sequences of operations.

Note that the routines "com1", "com2", and "com3" compute intermediate data that are common between the rigid-body and flexible-body computations for bodies 1, 2, and 3, respectively. Since the amount of computation involved in these routines is relatively small compared to that in other parts of the code, it was concluded that computing them once and communicating the results would take longer than repeating the computations. Therefore, these computations were repeated in appropriate processors rather than being distributed. This is indicative of the care that must be taken to minimize the sequential part of the overall computation in concurrent processing as implied by Amdahl's Law cited above.

Also, note that a distributed block Successive Over-Relaxation (SOR) algorithm (e.g. Reference 5) was used to solve the simultaneous linear equations, $A \cdot u_d = b$, for u_d at each time step. For the Space Station simulation on CAPPS, the SOR algorithm is more advantageous than L-U or other direct decomposition algorithms. There are 3 major advantages. First, while SOR is iterative, the solution from the previous time step is an effective starting guess to the solution at the current time step. Second, since the iterative algorithm is self-adaptive to variations in the computational load and the average number of SOR iterations decreases as the simulation progresses, the SOR algorithm is actually more efficient than L-U decomposition. And third, the communication pattern among processors is simple and allows high performance to be achieved on CAPPS.

Finally, the load distribution just discussed for the Space Station (Figure 3) was done by extensively editing the FORTRAN equations generated by SYMBOD. Editing the FORTRAN was a laborious but one-time experience. This experience taught us how the process can be imbedded in the SYMBOD code in a generalized form, a task left for future implementation.

4.0 SIMULATION PERFORMANCE AND RESULTS ON CAPPS

To demonstrate the CAPPS, a transient maneuver of the Space Station was simulated. The maneuver involved 10 degree rotations of both solar arrays about the ALPHA gimbals. The maneuver represents reorienting and then controlling the solar arrays to be perpendicular to the sun line. The control system executes the solar-array maneuver and simultaneously acts to maintain the central body of the Space Station in a fixed attitude with one axis pointing along the local vertical, and a plane containing that axis pointing along the velocity vector. Starting with quiescent initial conditions and no external disturbances, the control systems were turned on at time $t=0$ and the maneuver was terminated after simulating 200 seconds.

Simulation results and execution times were obtained on 1 and 8 CAPPS processors as well as on a SUN workstation and an IBM 3090 supercomputer (see Table 2). The IBM 3090 was chosen for comparison here because in prior benchmarks conducted by TRW, using a comparative simulation problem, the IBM 3090 throughput rate exceeded those of the Cray XMP, Cray 1S, Cray 2, and CYBER 205 supercomputers by 5, 17, 74, and 162 percent, respectively.

Table 2 contains both the CPU times for the 200 second simulated maneuver and the corresponding ratios of CPU time to real time. The 1-processor CAPPS, SUN workstation, and IBM-3090 all ran the same sequential code. The 8-processor CAPPS ran the parallelized version of the same simulation code. The simulations were performed with a fixed integration time step of 0.005 seconds, which was dictated by the highest frequency (10 Hz) present in the differential equations of motion. The

8-processor CAPPS simulation is a factor of 5.61 times faster than the 1-processor version, indicating an overall efficiency of 70.4 percent.

Execution times for the "coarse-grain" balanced computational load distribution among CAPPS' 8 processors are shown in Figure 4. the computational elements shown in the figure correspond to those shown in Figure 3 of Section 3. Note the idle times in the distributed load of each of the processors. The largest idle time was in CU4, which was not allocated any dof. Also note that roughly 40% of the total computation time was spent in the SOR solution and numerical integration.

It is interesting to consider in more detail the SOR linear equation solution part of the simulation. The algorithm is similar to block SOR (Reference 5), but it was specially tailored to the CAPPS and Space Station simulation. The distributed algorithm was run on the CAPPS with 1, 2, 4, and 8 processors and with different size matrices representing multibody systems of different numbers of dof. The execution times are presented in Figure 5, where the speedup factor is plotted against the number of processors with the computational load as a parameter. The speedup factor is the ratio of computational time with 1 processor to that with m processors solving the same, fixed size problem. Since memory size of the prototype CAPPS used limited the largest matrix that could be held by 1 processor to approximately $n=500$, the speedup factors for large problems are scaled factors as discussed in Reference 6.

A significant conclusion based on the results of Figure 5 is that the efficiency (defined as the speedup factor divided by the number of processors) of the CAPPS increases sharply as a function of the computational load. As the latter increased from 72 to 1200 dof, the 8-processor system's efficiency increased from 40 to 92 percent. This behavior of a loosely coupled concurrent processing system is explained by the observation that, to a first approximation, the parallel parts of the problem scale with the problem size, whereas the non-parallel parts (including communication) do not. As the problem size increases, the non-parallel operations constitute a smaller percentage of the total computational load.

Finally, Figure 6 contains 4 temporal plots of representative state vector entries. They are: a) the relative angular rotation of the starboard ALPHA gimbal, b) the first time derivative of the relative angular rotation of the starboard ALPHA gimbal, c) the inertial angular velocity of the central body along the 1 axis, and d) the fourth elastic displacement function of the starboard body. Comparing the ALPHA gimbal rotation and rotation rate plots, one can see evidence of flexible motion superposed on the rigid-body motion at the beginning of the maneuver. Also, one can see evidence in the

elastic displacement function shown that the bending deformation of the solar arrays is fully coupled to the rigid-body motion of the system. While only 4 plots are presented here, all entries of the state vector and its first time derivative as obtained from the four simulations were compared and found indistinguishable.

5.0 CONCLUSIONS

This work represents a major advance in the state of the art for analytical simulation of large space systems. Concurrent processing now offers the capability of simulating very large and complex mathematical models of multibody dynamical systems at high speeds and at an acceptable cost.

The performance to cost ratio of loosely coupled concurrent processors (CAPPS) vis-a-vis sequential computers was demonstrated to increase with computational load.

Having an explicit mathematical model is invaluable for "coarse-grain" computational load distribution, balancing, tuning, and otherwise maximizing the simulation throughput rate. The Symbol Manipulation Program (SMP) conveniently generates the explicit model.

The simulation process is divided into model development, computational load distribution, and computational load balancing steps. For practical application, all three steps must be mechanized to render most of the explicit model generation and load balancing process transparent to the user. This is feasible, based on the experiences reported herein.

Finally, on going work endeavors to incorporate an n-order algorithm for multibody equations together with explicit modeling and concurrent processing. Preliminary results, not reported here, demonstrates that this provides the capability of simulating, in real time, multibody systems with hundreds of large motion degrees of freedom.

ACKNOWLEDGEMENTS

The authors wish to express their sincere appreciation to Dr. Sam Lee of TRW Space & Technology Group for his development of the explicit mathematical model of the Space Station, and to Dr. Paul Cooper, Tom Sutter and John Young of NASA/LaRC and Jalal Mapar of Grumman Aerospace Corp. for supplying the "raw" data necessary for this development. Special thanks are due Dr. Kenneth Cox of NASA/JSC for his encouragement and support throughout the Space Station simulation development program.

REFERENCES

1. _____, "A Custom architected Parallel Processing System for the Space Station," TRW No. EML-003, Final report, Contract No. NAS9-17778, NASA Lyndon B. Johnson Space Center, May 1989.
2. Bodley, C.S., Devers, A.D., Park, A.C., and Frisch, H.P., "A Digital Computer Program for Dynamic Interaction Simulation of Controls and Structures (DISCOS)," Technical Paper 1219, Vols. 1 and 2, NASA, May 1978.
3. Singh, R.P., VanderVoort, R.J., and Likins, P.W., "Dynamics of Flexible Bodies in Tree Topology -- A Computer Oriented Approach," Paper No. AIAA-84-1024, AIAA/ASME/ASCE 25th Structures, Structural Dynamics, and Materials Conference, Palm Springs, CA. May 14-18, 1984.
4. Lee, S.S., "Symbolic Generation of Equations of Motion for Dynamics/Control Simulation of Large Flexible Multibody Space Systems," Ph.D. Dissertation, University of California, Los Angeles, 1988.
5. J. M. Ortega, Introduction to Parallel and Vector Solution of Linear Systems, Plenum Press, New York, 1988.
6. Gustafson, J.L., et al., "Development of Parallel Methods for a 1024-Processor hypercube," SIAM Journal of Scientific and Statistical Computations, Vol. 9, No. 4, July 1988.

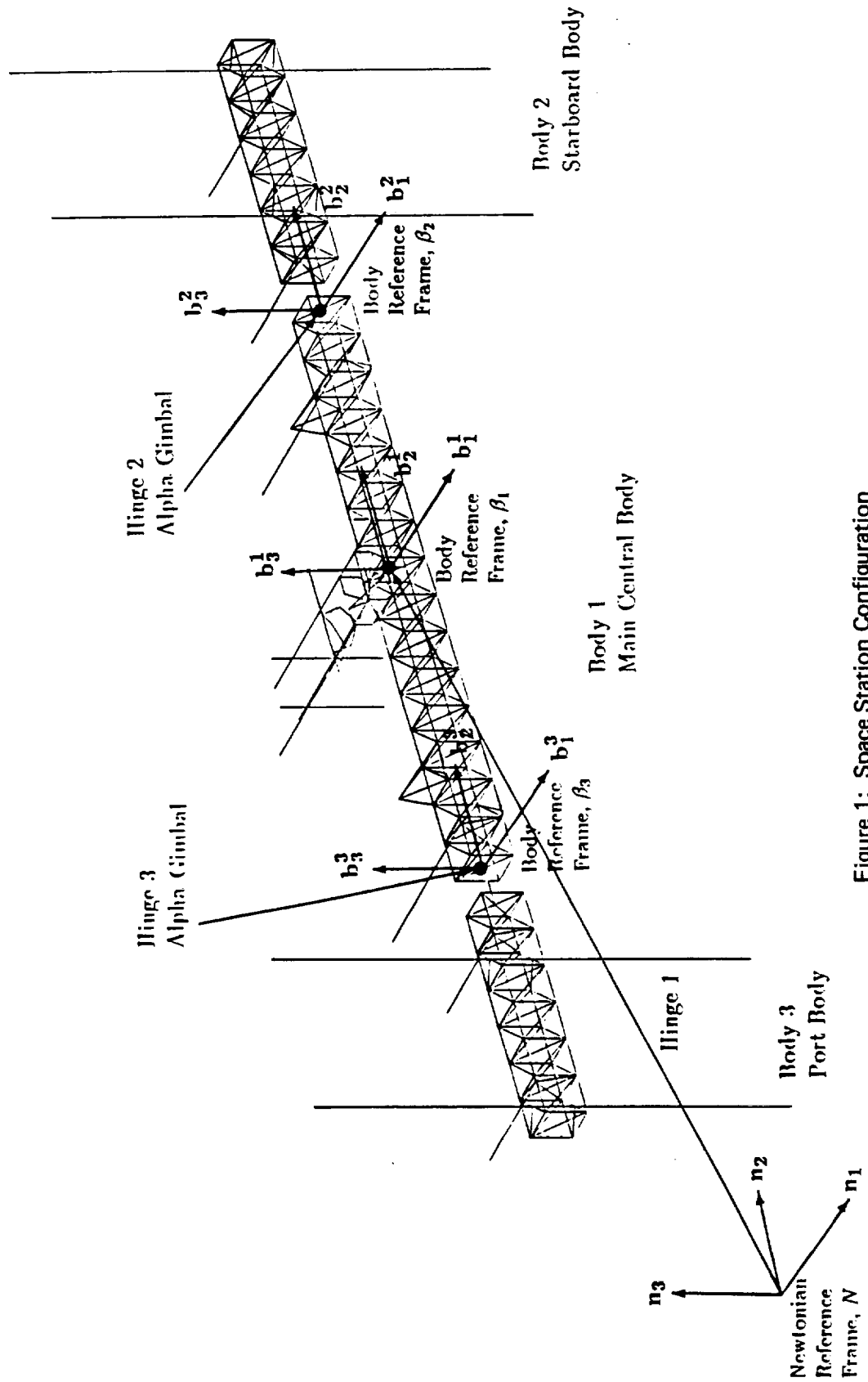


Figure 1: Space Station Configuration

Table 1: Space Station Model and Mass Properties Data

Model	Central Body	Starboard Body	Port Body
Finite Element Models:			
Grids	160	72	72
Elements	315	120	120
DOF	942	270	270
Mass and Inertia Data:			
Mass data (lb)			
Mass	373786	26685	26685
Center of mass (in) ^a			
X ₁	0.0	-29.4	-29.4
X ₂	0.0	733.3	-733.3
X ₃	0.0	16.4	16.4
Centroidal Inertia data (lb · in ²) ^b			
I ₁₁	8.047E10	6.973E09	6.973E09
I ₂₂	6.749E10	3.162E09	3.163E09
I ₃₃	1.114E11	4.836E09	4.836E09
I ₁₂	9.092E08	3.408E07	-3.557E07
I ₁₃	-5.099E09	1.243E07	1.243E07
I ₂₃	3.296E09	2.292E07	-2.441E07

^ameasured from origin of β reference frame

^bat CM about β reference frame axes

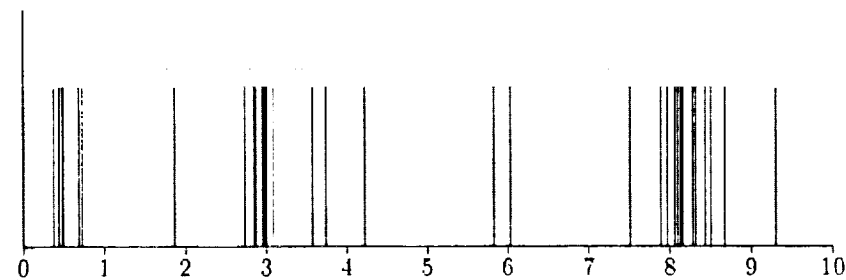
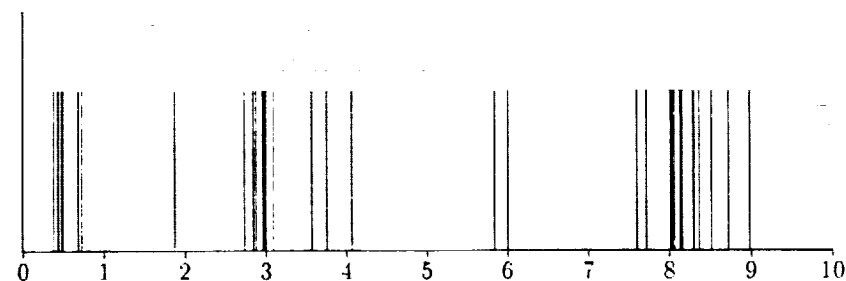
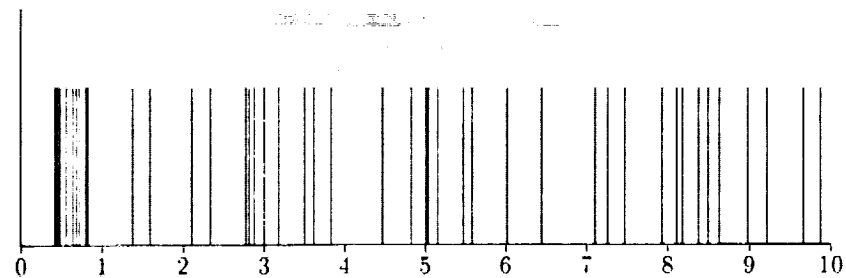


Figure 2: Frequency Spectra of the Space Station Model

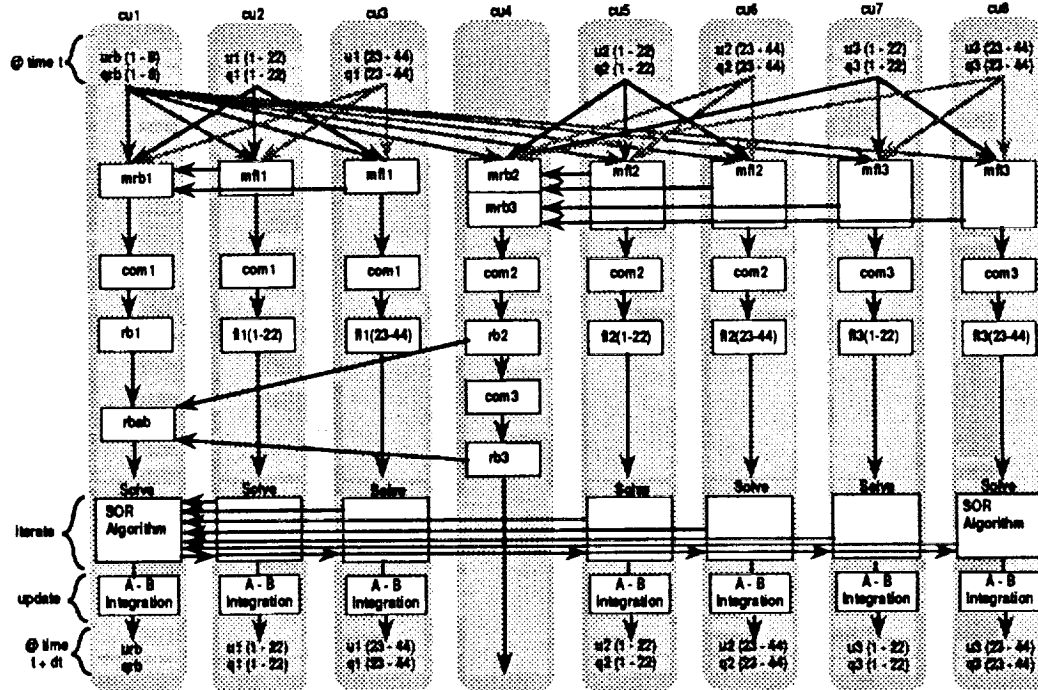


Figure 3: Computational Load Distribution for the Space Station Simulation on CAPPS

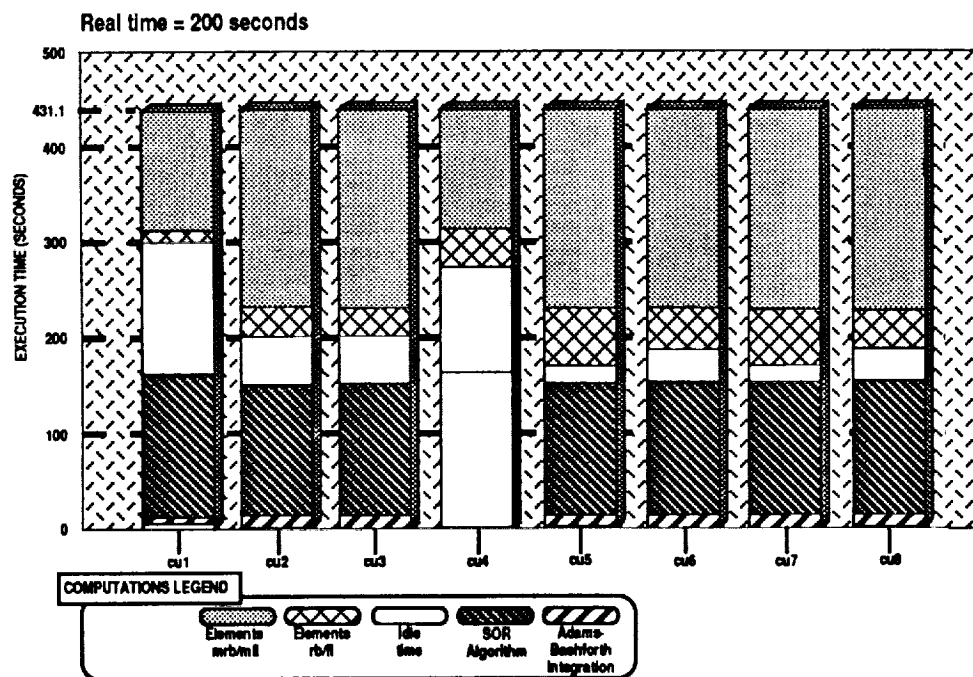


Figure 4: Execution Time for Coarse-Grain Balanced Computational Load

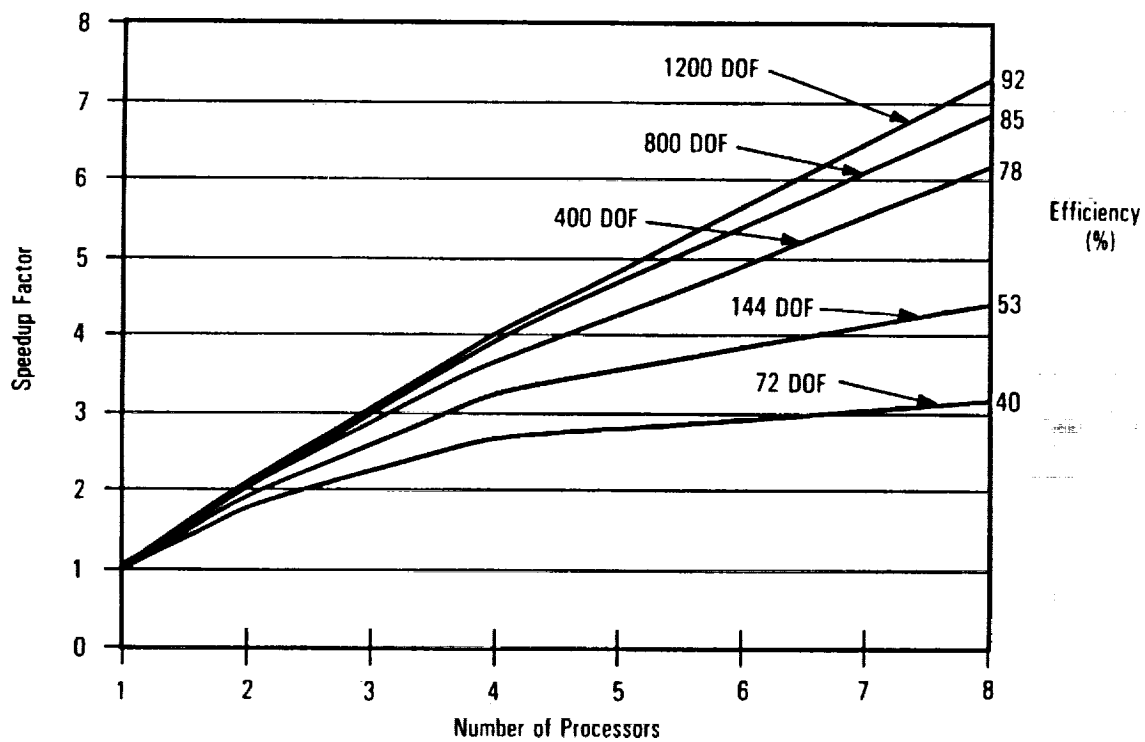


Figure 5: Speedup Factors for the Successive Over Relaxation Algorithm on CAPPS B-32

Table 2: Space Station Simulation Results

PARAMETER	CAPPS B - 32		IBM 3090/180E	SUN 25MHz
	1 CU	8 CUS**		
CPU TIME (MINUTES)	40.3	7.2	28.2	1844.8
CPU TIME/REAL TIME *	12.1	2.2	8.5	553.4

* Real time simulation - 200 seconds

** 8 - CU CAPPS speedup factor: 5.6 (70 percent overall efficiency)

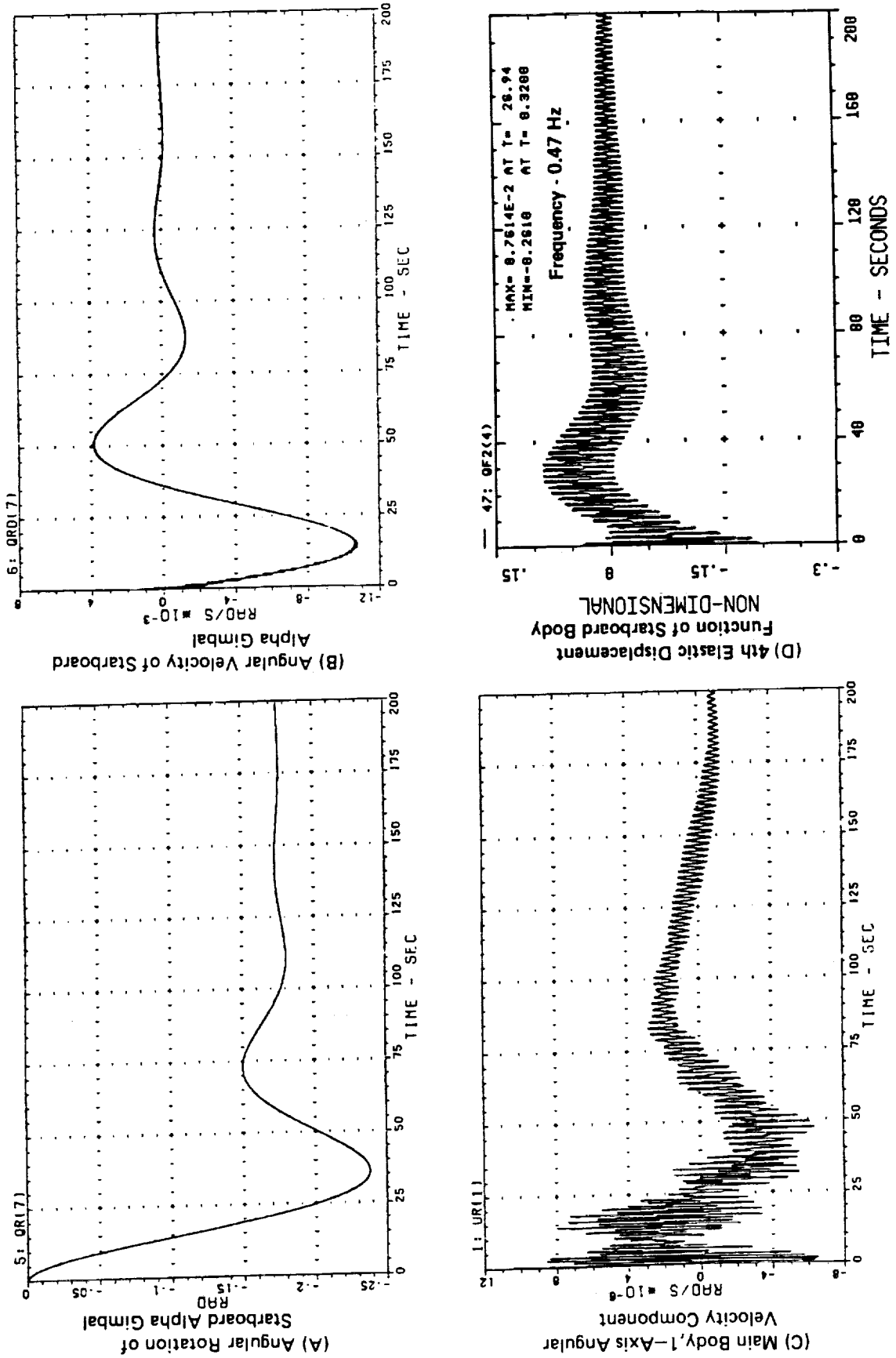


Figure 6: Representative Time Histories of the Flexible Space Station